APPENDIX
to Supplemental Amendment
for Application No. 10/709,360

United States Court of Appeals for the Federal Circuit

2007-1249, -1286

MICROPROCESSOR ENHANCEMENT CORPORATION
and MICHAEL H. BRANIGIN,

Plaintiffs-Appellants,

v.

TEXAS INSTRUMENTS INCORPORATED,

Defendant-Appellee,

and

INTEL CORPORATION,

Defendant-Appellee.

Lawrence M. Hadley, Hennigan, Bennett & Dorman, LLP, of Los Angeles, California, argued for plaintiffs-appellants. With him on the brief was Roderick G. Dorman. Of counsel were Mieke Katherine Malmberg and Omer Salik.

Gary N. Frischling, Irell & Manella LLP, of Los Angeles, California, argued for defendant-appellee Texas Instruments Incorporated. With him on the brief were Joseph M. Lipner, Brian D. Ledahl, Keith A. Orso, and Alexander L. Karpman.

Chad S. Campbell, Perkins Coie Brown & Bain P.A, of Phoenix, Arizona, argued for defendant-appellee Intel Corporation. With him on the brief were Mark E. Strickland and Aaron Matz. Of counsel on the brief was Tina M. Chappell, Intel Corporation, of Chandler, Arizona.

Appealed from: United States District Court for the Central District of California

Chief Judge Alicemarie H. Stotler

United States Court of Appeals for the Federal Circuit

2007-1249, -1286

MICROPROCESSOR ENHANCEMENT CORPORATION
and MICHAEL H. BRANIGIN,

Plaintiffs-Appellants,

v.

TEXAS INSTRUMENTS INCORPORATED,

Defendant-Appellee,

and

INTEL CORPORATION,

Defendant-Appellee.

Consolidated appeal from the United States District Court for the Central District of California in case nos. 05-CV-00323 and 05-CV-05667, Chief Judge Alicemarie H. Stotler.

DECIDED: April 1, 2008

Before NEWMAN, GAJARSA, and DYK, Circuit Judges.

GAJARSA, Circuit Judge.

This is a patent infringement case. Microprocessor Enhancement Corporation and Michael H. Branigin (collectively "MEC") appeal the judgments of the United States District Court for the Central District of California, Docket Nos. 05-CV-00323 and 05-CV-05667, wherein the district court found on summary judgment that Texas Instruments Incorporated ("TI") and Intel Corporation ("Intel") did not infringe any claim of U.S. Patent No. 5,471,593 ("the '593 patent") owned by MEC and that all claims of the patent

are invalid for indefiniteness. Because the district court erroneously concluded that the claims are indefinite, we reverse the court's finding of invalidity. Because the district court correctly construed the term "pipeline stage," we affirm the court's judgment of noninfringement.

BACKGROUND

The '593 patent is directed to computer processor architecture and methods for increasing microprocessor efficiency.[1] A computer program is composed of thousands to millions of instructions, which are stored in a computer's random access memory ("RAM"). Microprocessors implement programs by performing the operations specified by the instructions. To execute an instruction, a microprocessor must perform a series of tasks, and each task is completed on a fixed time interval defined by the system clock—a clock cycle. The tasks necessary to execute an instruction may be described generally as follows: (1) fetch—the processor gets the instruction from RAM; (2) decode—the processor reads and interprets the instruction; (3) issue—the processor sends the instruction to the appropriate functional unit; (4) execute—the functional unit executes the operation specified by the instruction; and (5) write—the result of the instruction is written to memory. In a most basic architecture, the entire microprocessor can be devoted to the sequential performance of these steps, such that the results of a complete instruction can be written to memory at a rate of one instruction per five clock cycles.

---

[1] We note that this is a general discussion of the relevant technology and the patent sufficient to introduce the concepts necessary for our analysis. Our legal conclusions herein are not premised on an assumption that this general discussion is a complete description of relevant technology.

Pipelined processors, however, operate like assembly lines, where the processor is subdivided into segments, each of which simultaneously completes its respective task on a different instruction. Encyclopedia of Computer Science and Engineering 1143 (Anthony Ralston ed., 2d ed. 1983); David A. Patterson & John L. Hennessy, Computer Architecture a Quantitative Approach 251 (1990). A pipelined processor is thus analogous to an assembly line designed to fetch a new instruction from memory before the previous instruction is completed and written to memory. For a linear set of instructions (i.e., a set of instructions that are neither branched nor conditional, discussed infra), a pipelined processor operates at maximum efficiency where one instruction is completed and one instruction is fetched on every clock cycle once the pipeline is full.

In order to operate in a useful fashion, programs often require the use of nonlinear instructions, i.e., instructions containing a branch or discontinuity in the instructional sequence, that result in "dependencies" among the individual instructions of an instruction set. Control dependencies occur, for example, when an instruction cannot be executed until the result of a prior conditional branch instruction is known. That is, a conditional instruction may specify that subsequent instructions are to be fetched and executed out of sequence, depending on whether a particular condition is satisfied. '593 patent col.2 ll.30–35.

The '593 patent labels one prior art method of processing this type of dependency as "conditional issuance." Id. at col.21 ll.42–66. Conditional issuance modifies the architecture of a pipelined processor by including a new segment called the conditional execution decision logic ("CEDL"). When a conditional instruction is

detected by the CEDL, the CEDL "locks" the issue segment to prevent the issuance of further instructions into the functional unit, until it can determine if the condition is satisfied. For every clock cycle during which the conditional instruction is held in the issue unit while the condition is determined, a "hole" is inserted into the pipeline at the unit immediately following the issue unit—i.e., one or more subsequent units of the pipeline will be nonoperational while waiting for the next issued instruction. If the condition is satisfied, the CEDL allows the conditional instruction depending on that condition to issue into the functional unit. If the condition is not satisfied, all conditional instructions depending on that condition and currently waiting in the pipeline are discarded, and subsequent instructions are fetched from memory. In the latter scenario, an additional number of holes equal to the number of discarded instructions are inserted into the pipeline.

The '593 patent describes and claims "conditional execution" as an improvement to conditional issuance. Id. at col.20 ll.13–18. Rather than controlling the issuance of the conditional instruction to the functional unit, the '593 patent teaches that the CEDL should be moved into the functional unit to control whether the results of a conditional instruction that has been executed will be written to memory. Id. Accordingly, when the CEDL detects a conditional instruction, it locks the execute segment to prevent the results of an executed conditional instruction from being forwarded to the write unit until the CEDL determines whether the condition is satisfied. Id. at cols.21–22. In this fashion, conditional execution may insert fewer holes into the pipeline than conditional issuance while a condition code is being determined. Id. at col.13 ll.46–48.

Independent claim 1 is a method claim and states as follows:

> 1. A method of executing instructions in a pipelined processor comprising:
>
> a conditional execution decision logic pipeline stage and at least one instruction execution pipeline stage prior to said conditional execution decision logic pipeline stage;
>
> at least one condition code;
>
> said instructions including branch instructions and non-branch instructions and each instruction including opcodes[2] specifying operations, operand specifiers specifying operands,[3] and conditional execution specifiers;
>
> said pipelined processor further including at least one write pipeline stage for writing the result(s) of each instruction to specified destination(s);
>
> at least one of the instructions including a means for specifying writing said condition code with a condition code result;
>
> the conditional execution decision logic pipeline stage performing a boolean algebraic evaluation of the condition code and said conditional execution specifier and producing an enable-write with at least two states, true and false; and
>
> said enable-write when true enabling and when those [sic] disabling the writing of instruction results at said write pipeline stage;
>
> said method further comprising the steps of:
>
> fetching source operands specified by said operand specifiers;
>
> performing the operation specified by said opcode;
>
> fetching the condition code, when specified by the conditional execution specifier, at the pipeline stage immediately preceding the conditional execution decision logic pipeline stage;
>
> operating the conditional execution decision logic pipeline

---

[2] Opcodes are fields in the instruction that specify the operation to be performed in the processor, commonly "Add," "Subtract," "Multiply", "Divide," "Compare," "Load," "Store," etc.

[3] Operands are the data to be operated on. Operand specifiers are fields in the instruction that specify the location of the operands.

stage, when specified by the conditional execution specifier, to determine the enable-write using the boolean algebraic evaluation;

writing said non-branch instruction results to a destination specified by the operand specifiers of the executing instruction and writing condition code results to the condition code when specified by the operand specifiers of the executing instruction, if the enable write is true; and

discarding or not writing the non-branch instruction results and discarding or not writing the condition code, if the enable-write is false.

'593 patent col.129 l.26 to col.130 l.33.

Independent claim 7 is an apparatus claim and states as follows:

7. A pipelined processor for executing instructions comprising:

a conditional execution decision logic pipeline stage, a[t] least one instruction execution pipeline stage prior to said conditional execution decision logic pipeline stage;

at least one condition code;

said instructions including branch instructions and non-branch instructions and including opcodes specifying operations, operand specifiers specifying operands, and conditional execution specifiers;

the pipelined processor further including at least one write pipeline stage for writing the result(s) of each instruction to specified destination(s);

at least one of the instructions including a means for specifying writing said condition code with a condition code result;

the conditional execution decision logic pipeline stage performing a boolean algebraic evaluation of the condition code and said conditional execution specifier and producing an enable-write with at least two states, true and false;

said enable-write when true enabling and when false disabling the writing of instruction results at said write pipeline stage;

fetching means for fetching source operands specified by said operand specifiers;

> operating means for performing the operation specified by said opcode;
>
> condition code fetching means for fetching the condition code, when specified by the conditional execution specifier, at the pipeline stage immediately preceding the conditional execution decision logic;
>
> the conditional execution decision logic pipeline stage, when specified by the conditional execution specifier, determining the enable-write using the boolean algebraic evaluation;
>
> writing means for writing said non-branch instruction results to a destination specified by the operand specifiers and writing to the condition code when specified, if enable-write is true; and
>
> said writing means further for discarding or not writing the non-branch instruction results and discarding or not writing the condition code, if enable-write is false.

'593 patent col.131 l.13 to col.132 l.3.

Initially, MEC filed a single suit against both TI and Intel, alleging that TI's C6000 digital signal processor and Intel's Itanium 2 microprocessors infringed claims 1, 5, 7, and 11 of the '593 patent. The parties, however, concluded that Intel had been misjoined and stipulated to the dismissal of MEC's claims against Intel without prejudice. MEC subsequently refiled its claims against Intel in a separate suit, but moved to consolidate discovery in the two cases. The court denied the motion.

In the TI case, the court issued two separate opinions concluding that TI's motions for summary judgment of invalidity and noninfringement would be granted. Microprocessor Enhancement Corp. v. Tex. Instruments Inc., No. SA CV 05-323, 2007 WL 840362 (C.D. Cal. Feb. 8, 2007) ("Invalidity"); Microprocessor Enhancement Corp. v. Tex. Instruments Inc., No. SA CV 05-323, 2007 WL 840364 (C.D. Cal. Feb. 8, 2007) ("Noninfringement"). Pursuant to Central District of California Local Rule 56-1, the court's opinions contained a "statement of the facts which are uncontroverted or as to

which there is no substantial controversy as well as the conclusions of law that follow therefrom."[4] As provided in Central District of California Local Rule 56-3, the court based this statement on the proposed "Statement of Uncontroverted Facts and Conclusions of Law" submitted by TI. On February 8, 2007, the court entered a take nothing judgment in TI's favor. Microprocessor Enhancement Corp. v. Tex. Instruments Inc., No. SA CV 05-323, 2007 WL 840367 (C.D. Cal. Feb. 8, 2007).

At the time the trial court entered judgment in the TI case, cross motions for summary judgment were pending in the Intel case. In particular, Intel had moved for summary judgment of noninfringement. Rather than wait for the court to rule on the motions, however, MEC and Intel filed a stipulated final adjudication of their case. The stipulated adjudication explicitly recognized that MEC would be collaterally estopped from challenging the invalidity ruling of the TI case and that the court would apply the claim construction of the TI case, under which Intel's accused products would not infringe any claims of the '593 patent. As part of the stipulated dismissal, MEC and Intel also agreed to file a "Joint Submission of Additional Evidence," which included evidence of the type that would have been submitted in opposition to the summary judgment briefs already filed. The stipulated adjudication contained a proposed order adopting the parties' stipulations and incorporating the Joint Submission of Additional Evidence

---

[4] Central District of California Local Rule 56-1 provides that a movant for summary judgment shall include "a proposed 'Statement of Uncontroverted Facts and Conclusions of Law' and the proposed judgment. Such proposed statement shall set forth the material facts as to which the moving party contends there is no genuine issue." When deciding the motion for summary judgment, the court assumes that the facts contained in the proposed statement are "admitted without controversy" unless they are included in the "Statement of Genuine Issues" (required of the nonmoving party pursuant to C.D. Cal. Local Rule 56-2) and controverted by declaration or written evidence. C.D. Cal. Local Rule 56-3.

into the record of the Intel case, which the court signed and entered on March 8, 2007. Accordingly, the court entered a take nothing judgment in Intel's favor.

MEC filed its notice of appeal in the TI case on March 7, 2007 and its notice of appeal in the Intel case on March 26, 2007. On May 23, 2007, MEC filed a motion to consolidate the appeals, and TI's response was therefore due on June 4, 2007. Fed. R. App. P. 26(a), 27(a)(3)(A). Nevertheless, the clerk granted the motion to consolidate on May 30, 2007, before TI filed a response. We have jurisdiction over the appeal pursuant to 28 U.S.C. § 1295(a).

DISCUSSION

At the outset, the parties dispute the scope of the record on appeal. MEC argues that the case was properly consolidated and that the record therefore includes evidence submitted in the Intel case, including the Joint Submission of Additional Evidence. TI counters that because MEC admitted to being collaterally estopped in the Intel case from challenging the invalidity ruling of the TI case, and because MEC stipulated to the claim constructions rendered in the TI case, this court should only consider the evidence presented in the TI case below. In support of its position, TI argues that "[e]vidence that was not before the district court at the time of the summary judgment proceeding . . . cannot be invoked to challenge the summary judgment order." L&W, Inc. v. Shertech, Inc., 471 F.3d 1311, 1315 n.2 (Fed. Cir. 2006).

Although L&W's statement as to the scope of the record on appeal is well-supported as a general matter of law, cf. Monarch Knitting Mach. Corp. v. Sulzer Morat GmbH, 139 F.3d 877, 880 (Fed. Cir. 1998) (evidence submitted to a district court after entry of final judgment under Federal Rule of Civil Procedure 54(b) not part of record on

appeal from that judgment); Laitram Corp. v. Cambridge Wire Cloth Co., 919 F.2d 1579, 1581 & n.4 (Fed. Cir. 1990) (district court properly excluded from record on appeal, those exhibits not before it when summary judgment was entered); cf. also Kirshner v. Uniden Corp., 842 F.2d 1074, 1077 (9th Cir. 1988) (evidence not admitted by the district court cannot be part of the record on appeal); Fassett v. Delta Kappa Epsilon, 807 F.2d 1150, 1165 (3d Cir. 1986) (district court not authorized to augment record on appeal with evidence not on record at the time it rendered final decision), we are unable to locate any authority addressing the scope of the appellate record when the trial record differs for the cases in a consolidated appeal. We need not, however, decide whether the statement of law in L&W governs the scope of the record in this consolidated appeal. Because extrinsic evidence is "'less significant than the intrinsic record in determining the legally operative meaning of claim language,'" Phillips v. AWH Corp., 415 F.3d 1303, 1317 (Fed. Cir. 2005) (en banc) (additional internal quotations omitted) (quoting C.R. Bard, Inc. v. U.S. Surgical Corp., 388 F.3d 858, 862 (Fed. Cir. 2004)), our decision herein rests primarily on the intrinsic record of the '593 patent. In addition, any differences between the extrinsic record developed in the two cases below does not contradict our reading of this intrinsic record. We therefore do not decide the precise demarcation between that evidence which is properly before us and that which is not.

I. INVALIDITY

The district court concluded that independent claims 1 and 7 of the '593 patent are invalid for indefiniteness on the grounds that both claims impermissibly mix two distinct classes of patentable subject matter and that the claims are insolubly ambiguous for requiring that a single word be interpreted differently in different portions

of a single claim. Invalidity, 2007 WL 840362, at *2–*4. Under 35 U.S.C. § 112, ¶ 2, the claims of a patent must "particularly point[ ] out and distinctly claim[ ] the subject matter which the applicant regards as his invention." "A claim is considered indefinite if it does not reasonably apprise those skilled in the art of its scope." IPXL Holdings, L.L.C. v. Amazon.com, Inc., 430 F.3d 1377, 1383–84 (Fed. Cir. 2005). "Because a claim is presumed valid, a claim is indefinite only if the 'claim is insolubly ambiguous, and no narrowing construction can properly be adopted.'" Honeywell Int'l, Inc. v. Int'l Trade Comm'n, 341 F.3d 1332, 1338–39 (Fed. Cir. 2003) (quoting Exxon Research & Eng'g Co. v. United States, 265 F.3d 1371, 1375 (Fed. Cir. 2001)). Whether a claim reasonably apprises those skilled in the art of its scope is a question of law that we review de novo. Exxon Research, 265 F.3d at 1376 ("[D]etermination of claim indefiniteness is a legal conclusion that is drawn from the court's performance of its duty as the construer of patent claims."). We turn to each of the district court's indefiniteness rulings in turn.

First, we conclude that neither claim 1 nor claim 7 impermissibly claim mixed classes of subject matter. A single patent may include claims directed to one or more of the classes of patentable subject matter, but no single claim may cover more than one subject matter class. IPXL Holdings, 430 F.3d at 1384 (holding indefinite a claim covering both an apparatus and a method of using that apparatus). Applying this rule, the district court concluded that although claim 1 purported to claim a method of executing instructions in a pipelined processor, the structural limitations of the pipelined processor evidence an intent to claim the apparatus as well. Invalidity, 2007 WL 840362, at *3. The court similarly concluded that although claim 7 purported to be an

apparatus claim, the functional limitations are directed to the use of the apparatus rather than functional descriptions of certain claimed features of the apparatus. We disagree.

The drafting structure of claim 1 may be generally described as follows:

> 1. A method of executing instructions in a pipelined processor comprising:
>
> [structural limitations of the pipelined processor];
>
> the method further comprising:
>
> [method steps implemented in the pipelined processor].

See '593 patent col.129 l.27 to col.130 l.32. Although this seeming preamble within a preamble structure is unconventional, its effect on the definiteness of claim 1 lacks the conclusiveness with which King Claudius's guilt is established by his reaction to Hamlet's play within a play. See William Shakespeare, Hamlet act 3, sc. 2. Method claim preambles often recite the physical structures of a system in which the claimed method is practiced, and claim 1 is no different. The conclusion of IPXL Holdings was based on the lack of clarity as to when the mixed subject matter claim would be infringed. 430 F.3d at 1384 ("[I]t is unclear whether infringement of claim 25 occurs when one creates a system that allows the user to [practice the claimed method step], or whether infringement occurs when the user actually [practices the method step]."). There is no similar ambiguity in claim 1 of the '593 patent. Direct infringement of claim 1 is clearly limited to practicing the claimed method in a pipelined processor possessing the requisite structure.

In similar fashion, claim 7 does not cover both an apparatus and a method of use of that apparatus. As this court recently stated, apparatus claims are not necessarily indefinite for using functional language. See Halliburton Energy Servs. v. M-I LLC, 514 F.3d 1244, 1255 (Fed. Cir. 2008). Indeed, functional language in a means-plus-function

format is explicitly authorized by statute. 35 U.S.C. § 112, ¶ 6. Functional language may also be employed to limit the claims without using the means-plus-function format. E.g., K-2 Corp. v. Salomon S.A., 191 F.3d 1356, 1363 (Fed. Cir. 1999) (analyzing functional language as an additional limitation to an apparatus claim for an in-line skate). Moreover, where the claim uses functional language but recites insufficient structure, § 112, ¶ 6 may apply despite the lack of "means for" language. See, e.g., Personalized Media Commc'ns, LLC v. Int'l Trade Comm'n, 161 F.3d 696, 703–04 (Fed. Cir. 1998) (discussing cases). Notwithstanding these permissible instances, the use of functional language in a claim may "fail 'to provide a clear-cut indication of the scope of subject matter embraced by the claim' and thus can be indefinite." Halliburton, 514 F.3d at 1255 (quoting In re Swinehart, 439 F.2d 210, 212–13 (CCPA 1971)). Claim 7 of the '593 patent, however, is clearly limited to a pipelined processor possessing the recited structure and capable of performing the recited functions, and is thus not indefinite under IPXL Holdings.

Second, we conclude that neither claim 1 nor claim 7 is insolubly ambiguous in its use of the term "condition code." Claim 1 and claim 7 both claim "at least one condition code" as an element of the pipelined processor. Thereafter, claim 1 and claim 7 both contain five references to "the condition code" or "said condition code." The district court reasoned that where a subsequent use of a claim term makes reference to the first use as an antecedent by using "said" or "the," that term must be interpreted consistently across all such uses in a single claim. Invalidity, 2007 WL 840362, at *4 (citing Process Control Corp. v. HydReclaim Corp., 190 F.3d 1350, 1356–57 (Fed. Cir. 1999)). As used in claims 1 and 7, the term "condition code" must mean either a

storage unit or a value derived from the output of the storage unit depending on the context in which its used, yet both claims are facially nonsensical if either of these definitions is used exclusively. The district court applied its reading of Process Control, concluding that "condition code" must be construed consistently within a single claim and that the claims were therefore indefinite. Id.

Although we agree with the district court's initial assumption that a single "claim term should be construed consistently with its appearance in other places in the same claim or in other claims of the same patent," Rexnord Corp. v. Laitram Corp., 274 F.3d 1336, 1342 (Fed. Cir. 2001), the patentee's mere use of a term with an antecedent does not require that both terms have the same meaning. Specifically, Process Control did not announce a rule that the reference to an antecedent absolutely requires a term to be consistently construed across uses. Cf. Epcon Gas Sys., Inc. v. Bauer Compressors, Inc., 279 F.3d 1022, 1030–31 (Fed. Cir. 2002) ("'A word or phrase used consistently throughout a claim should be interpreted consistently.'" (quoting Phonometrics, Inc. v. Northern Telecom Inc., 133 F.3d 1459, 1465 (Fed.Cir.1998))).

Claim 1 at issue in Process Control reads as follows:

> A method of metering different material ingredients for discharge to a material processing machine, comprising:
>
> [a] delivering to a common hopper a plurality of individual material ingredients at controllable individual material discharge rates,
>
> [b] discharging material from said common hopper to said processing machine at a discharge rate,
>
> [c] determining loss of weight of material in said hopper due to discharge of material therefrom,
>
> [d] determining the material processing rate of the processing machine from the sum of the material discharge rates of the ingredients to the common hopper and the

> discharge rate of the material from the common hopper to the processing machine, and
>
> [e] controlling the material discharge rates of the ingredients to the common hopper in response to said determined material processing rate as needed to maintain a preset recipe of said blended ingredients at said determined material processing rate.

Process Control, 190 F.3d at 1354–55. The court did rule that "discharge rate" must be construed identically in limitations [b] and [d], but the court did not rely principally on antecedent basis to support its rationale.

> It is clear from the language of the claim itself that the term "a discharge rate" in clause [b] is referring to the same rate as the term "the discharge rate" in clause [d]. This conclusion necessarily results from the identical language associated with the term "discharge rate" in both clauses [b] and [d], namely "from the common hopper to the material processing machine."

Id. at 1356 (emphases added). The court then noted that "[i]n addition, [this] conclusion avoids any lack of antecedent basis problem for the occurrence of 'the discharge rate' in clause [d]." Id. at 1356–57. Given the well-settled rule that claims are not necessarily invalid for a lack of antecedent basis,[5] the court's observations regarding antecedent basis are merely supportive of, rather than necessary to, its conclusion that "discharge rate" must have a single consistent meaning in claim 1.

Turning to claim 1 and claim 7 of the '593 patent, we note that "[a] claim that is amenable to construction is not invalid on the ground of indefiniteness" if the construction renders the claim definite. Energizer Holdings, 435 F.3d at 1371. Unlike

---

[5] See, e.g., Energizer Holdings, Inc. v. Int'l Trade Comm'n, 435 F.3d 1366, 1370–71 (Fed. Cir. 2006) ("[D]espite the absence of explicit antecedent basis, '[i]f the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite.'" (quoting Bose Corp. v. JBL, Inc., 274 F.3d 1354, 1359 (Fed. Cir. 2001))).

the claim at issue in Process Control, "condition code" as used in claims 1 and 7 is not surrounded by uniform language that requires a single interpretation of the term. Cf. Epcon Gas Sys., 279 F.3d at 1031 (construing "substantially" as having two different meanings based on its use in "two contexts with a subtle but significant difference"). Rather, the appropriate meaning of "condition code" is readily apparent from each occurrence in context, and TI's expert, Dr. Patt, indicated that the '593 patent used condition code to refer to a value or a storage location based on its context within the claims. Indeed, the claims' apparent nonsensical reading under a uniform construction of "condition code" is indicative of the ease of determining the appropriate meaning of each use of the term from its context. For these reasons, the use of "condition code" in claim 1 and claim 7 does not render these claims indefinite.

II. NONINFRINGEMENT

The district court granted TI summary judgment of noninfringement on two separate bases. First, the district court construed the term "pipeline stage" to be "a structure that works on an instruction for a regular interval of time defined by the system clock (i.e., one or more clock cycles), with separate pipeline stages capable of simultaneously working on different instructions."[6] Infringement, 2007 WL 840364, at *3. Under this temporal construction, the claims require that the condition code be fetched during one clock cycle and used during the next clock cycle. E.g., '593 patent claim 1 ("fetching the condition code, when specified by the conditional execution specifier, at the pipeline stage immediately preceding the conditional execution decision

[6] For ease of reference, we refer to the district court's construction of "pipeline stage" as a "temporal" construction, inasmuch as it defines the term according to clock cycles.

logic pipeline stage"). Second, the district court construed the term "instruction execution pipeline stage" to be a "pipeline stage directed to performing the operation specified by the opcode of an instruction." Id. Under this definition, the claims require that the pipeline stage for performing opcode operations occur before the CEDL pipeline stage. E.g., '593 patent claim 1 ("at least one instruction execution pipeline stage prior to said conditional execution decision logic pipeline stage"). MEC admits that the accused products of both TI and Intel do not infringe if we affirm either of these two constructions. Accordingly, because we affirm the district court's construction of "pipeline stage," we need not address the construction of "instruction execution pipeline stage."

The term "pipeline stage" is used throughout claims 1 and 7. The term is usually used with a modifier that describes the function of the named pipeline stage, e.g., "conditional execution decision logic pipeline stage" or "instruction execution pipeline stage." Claim 1 and claim 7 both use "pipeline stage" without a modifier one time. For example, claim 1 reads, "fetching the condition code, when specified by the conditional execution specifier, at the pipeline stage immediately preceding the conditional execution decision logic pipeline stage." The district court applied its temporal construction of "pipeline stage" to both the modified and unmodified uses of the term in claims 1 and 7. Noninfringement, 2007 WL 840364, at *3–*4.

On appeal, MEC posits that the court's construction should not apply to unmodified uses of "pipeline stage," and argues that "the pipeline stage" indicates a structure at a particular position in the pipeline, rather than a structure that works with an instruction for one or more clock cycles. MEC does agree, however, that modified

uses of "pipeline stage," e.g., "instruction execution pipeline stage" and "CEDL pipeline stage," are temporal terms describing structures operating on complete clock cycles. Despite this admission, MEC nevertheless argues that the single unmodified use of "pipeline stage" in both claim 1 and claim 7 should be construed as a positional term.

We review the district court's claim construction de novo. Cybor Corp. v. FAS Techs., Inc., 138 F.3d 1448, 1456 (Fed. Cir. 1998) (en banc). Claim terms must be given "the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." Phillips, 415 F.3d at 1313. This court ascertains the meaning of a disputed term by looking to "'those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean.'" Id. at 1314 (quoting Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1116 (Fed. Cir. 2004)). "'Those sources include the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence . . . .'" Id. Phillips teaches that these sources should be accorded relative weights in the order listed, with the words of the claims themselves being the most relevant. Id. at 1314–19. Accordingly, we discuss each source of meaning of the claim term "the pipeline stage" in this order.

Beginning with the claims themselves, "the context in which a term is used in the asserted claim can be highly instructive." Id. at 1314. Because MEC agrees that modified uses of "pipeline stage" are temporal rather than positional terms, MEC necessarily agrees that the CEDL pipeline stage refers to a logic structure that utilizes the condition code during "a regular interval of time defined by the system clock." See Noninfringement, 2007 WL 840364, at *3 (construing "pipeline stage"). The use of the

term "pipeline stage immediately preceding" before the term "[CEDL] pipeline stage" therefore suggests that "pipeline stage immediately preceding" is itself a temporal rather than positional term. For this reason, to the extent that the term "the pipeline stage," is ambiguous as to whether it denotes time or position, the surrounding temporal language, including "CEDL pipeline stage" and "immediately preceding," suggests that "the pipeline stage" is also temporal. That is, construing "the pipeline stage" as a positional term seems inconsistent with the temporal context in which it is used.

We next turn to the specification, "'informed, as needed, by the prosecution history.'" Phillips, 415 F.3d at 1315 (quoting Multiform Desiccants, Inc. v. Medzam, 133 F.3d 1473, 1478 (Fed. Cir. 1998)). We note, however, that this is not a case identified by Phillips as one in which "the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges." Cf. id. at 1314. With regard to the specification, we are simply unable to discern whether a person of ordinary skill in the art would read "pipeline stage" as an exclusively temporal, rather than positional, term. For example, certain portions of the specification suggest the term has a temporal meaning: "Each stage in the pipeline, including the Reservation Stations, must determine if the pipeline will move forward on the next clock." '593 patent col.68 ll.61–63. Conversely, MEC notes that Figure 15a and various other figures lack sufficient structure to allow a condition code to be fetched during one clock cycle and stored until it is used by the CEDL pipeline stage during the clock cycle immediately following. MEC thus argues that the absence of storage structure indicates that the condition code must be used by the CEDL pipeline stage during the same clock cycle in which it is fetched, thereby requiring "the pipeline stage" to be a positional term

in the claims. Comparing these examples with the remainder of the specification, we are unable to determine whether a person of ordinary skill in the art would understand "the pipeline stage" to be used in a temporal or positional sense.

To resolve the ambiguity of the specification, we turn to the prosecution history. The term "the pipeline stage" was added by amendment during prosecution, and the amendment makes clear that the inventor intended that this sole unmodified use of "pipeline stage" would have the same temporal sense as the modified uses appearing throughout the claims. The examiner rejected the first independent claim then pending, claim 61, as indefinite. Office Action, '593 patent, at 2 (June 9, 1994). The relevant portions of claim 61 read as follows:

> A method of executing instructions in a pipelined processor:
>
> said pipelined processor including conditional execution decision logic and at least one condition code;
>
> . . .
>
> said method comprising the steps of:
>
> . . .
>
> fetching the condition code, when specified by the conditional execution specifier, at a pipeline position just preceding the conditional execution decision logic;
>
> operating the conditional execution decision logic, when specified by the conditional execution specifier, to determine, by said boolean algebraic evaluation, enabling and disabling of writing non-branch instruction results; and
>
> writing said non-branch instruction results to a destination specified by the operand specifiers and writing to the condition code when specified, if enabled by the conditional execution decision logic.

Amendment D, '593 patent, at Claims pp. 1–2 (Jan. 20, 1994). Rejected claim 61 did not use the term "pipeline stage" at all. Rather, it used the terms "conditional execution specifier" and "pipeline position," the latter of which the Examiner concluded was

indefinite because "it is not clear what the function of the pipeline position at that stage is." Office Action, '593 patent, at 2 (June 9, 1994) (emphasis added).

To address the rejection, the applicant made two amendments. First, the applicant amended the structure of the pipelined processor as follows: "said pipelined processor including a conditional execution decision logic pipeline stage, a[t] least one instruction execution pipeline stage prior to said conditional execution decision logic pipeline stage, and at least one condition code." Amendment A, '593 patent, at Claims pp.1–3 (Sept. 8, 1994). Second, the applicant amended the fetching limitation at issue on appeal as follows: "fetching the condition code . . . at ~~a pipeline position just~~ the pipeline stage immediately preceding the conditional execution decision logic pipeline stage."[7] Id. These amendments added a new pipeline stage with a specified function to the pipelined processor—the "at least one instruction execution pipeline stage"—and provided an antecedent basis for "the pipeline stage" (where the prior term "a pipeline position" neither had nor required an antecedent basis) in a way that specified its function. Accordingly, rather than intending that the unmodified use of "pipeline stage" denote position rather than time, this amendment indicates the applicant's intent that "the pipeline stage" take its antecedent basis, and thereby the function and temporal

[7] The applicant cancelled claim 61 and submitted new claim 73, but a comparison of original claim 61 and newly submitted claim 73 makes clear that claim 73 should be read as an amendment to claim 61, especially considering that claim 73 replaced claim 61 as the first independent claim.

meaning, from "at least one instruction execution pipeline stage."[8] No other reading of this amendment would address the examiner's indefiniteness rejection based on the indiscernible function of "a pipeline position."[9] We therefore conclude that the prosecution history of the term "the pipeline stage" is supportive intrinsic evidence that the inventor used the term "the pipeline stage" to refer to "a structure that works on an instruction for a regular interval of time defined by the system clock (i.e., one or more clock cycles), with separate pipeline stages capable of simultaneously working on different instructions."

In addition to the claim amendment inserting the term "the pipeline stage" into the claims of the '593 patent, the parent application, U.S. Application No. 07/448720 (filed Dec. 11, 1989, now abandoned), contained language more clearly evidencing an intent that the term "pipeline stage" be used in its temporal sense.

> To improve the clock rate . . . , most high performance architectures segment the functional units into several pieces called "pipeline stages." A single pipeline stage can be traversed in one clock cycle. With pipelining, each functional unit can be viewed as an assembly line capable of working on several instructions at different stages of completion . . . .

'720 application, at 4. This language was, however, removed by the applicant during

---

[8] As did this court in Process Control, we note that this construction of "the pipeline stage" avoids antecedent basis problems. The first (and only) unmodified use of "pipeline stage" is preceded by the definite article "the." Accordingly, the term "the pipeline stage" would properly take its antecedent basis from one of the previous uses of "pipeline stage," all of which are modified and thus denote structures that operate on one or more complete clock cycles. If, as suggested by MEC, the unmodified use of "pipeline stage" means something different, the first occurrence of the unmodified term should be "a pipeline stage."

[9] We note that this reading of the claim is not altered by further unrelated amendments during prosecution that resulted in claim 1 as issued.

prosecution of the '720 application in response to the examiner's statement that "elements/devices or groups of elements/devices which are conventional and generally widely known in the field of data processing ('DP') art should not be described in detail." Compare Office Action, '720 application, at 5 (Oct. 28, 1992) with Amendment B, '720 application, at Specification p. 4 (Jan. 23, 1993). Although this omitted language of the '720 application is not dispositive of our construction of "the pipeline stage" as used in the claims of the '593 patent, its probative value is twofold. First, this language is some evidence that the inventor used the term to denote structures delineated by clock cycles rather than position. Second, the omission of this language in response to the examiner's statement suggests that the inventor considered this clock cycle usage of "pipeline stage" to be "conventional and generally widely known in the field of data processing."

Lastly, having thoroughly examined "'the indisputable public records consisting of the claims, the specification and the prosecution history,'" Phillips, 415 F.3d at 1319 (quoting Southwall Techs., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1578 (Fed. Cir. 1995)), we find that the extrinsic evidence supports the conclusion that the inventor used "the pipeline stage" in its temporal sense, consistent with the term's ordinary meaning in the computer arts. For example, David A. Patterson & John L. Hennessy, Computer Architecture a Quantitative Approach 251 (1990), relied on by all three parties on appeal, discusses the concept of a pipe stage or pipe segment, and the discussion is framed by references to time and clock cycles rather than positions.

In sum, the district court correctly construed "pipeline stage," whether modified or standing alone, as "a structure that works on an instruction for a regular interval of time

defined by the system clock (i.e., one or more clock cycles), with separate pipeline stages capable of simultaneously working on different instructions." This construction is well supported by (1) MEC's admissions that modified uses of pipeline stage, e.g., "instruction execution pipeline stage" uses "pipeline stage" in the clock cycle sense of the word; (2) the structure and context of the term's use in the claims; (3) the prosecution history; and (4) the extrinsic evidence of how the term would be understood by a person of ordinary skill in the art. Because the parties agree that the accused products do not practice any of the asserted claims under this construction of "the pipeline stage," the district court correctly entered judgment of noninfringement in both the TI case and the Intel case.

CONCLUSION

Because we conclude that the asserted claims are not indefinite, the district court's judgment that the asserted claims of the '593 patent are invalid is reversed. Because we conclude that the district court correctly construed "pipeline stage," the district court's judgment of noninfringement is affirmed.

AFFIRMED-IN-PART, REVERSED-IN-PART

COSTS

No costs.